NASA Technical Memorandum 101704

A Knowledge-Based System Design/Information Tool for Aircraft Flight Control Systems

Dale A. Mackall and James G. Allen

October 1989

NASA
National Aeronautics and
Space Administration

NASA Technical Memoradum 101704

A Knowledge-Based System Design/Information Tool for Aircraft Flight Control Systems

Dale a Mackall
Ames Research Center, Dryden Flight Research Facility
Edwards, California

James G. Allen
Charles Stark Draper Laboratories
Cambridge, Massachusetts

1989

NASA
National Aeronautics and
Space Administration
Ames Research Center
Dryden Flight Research Facility
Edwards, California 93523-5000

A KNOWLEDGE-BASED SYSTEM DESIGN/INFORMATION TOOL FOR AIRCRAFT FLIGHT CONTROL SYSTEMS

Dale A. Mackall*
NASA Ames Research Center
Dryden Flight Research Facility
Edwards, California

and

James G. Allen†
Charles Stark Draper Laboratories
555 Technology Square
Cambridge, Massachusetts

Abstract

Research aircraft have become increasingly dependent on advanced electronic control systems to accomplish program goals. These aircraft are integrating multiple disciplines to improve performance and satisfy research objectives. This integration is being accomplished through electronic control systems. Because of the number of systems involved and the variety of engineering disciplines, systems design methods and information management have become essential to program success. The primary objective of the system design/information tool for aircraft flight control system is to help transfer flight control system design knowledge to the flight test community. By providing all of the design information and covering multiple disciplines in a structured, graphical manner, flight control systems can more easily be understood by the test engineers. This will provide the engineers with the information needed to thoroughly ground test the system and thereby reduce the likelihood of serious design errors surfacing in flight. The secondary objective is to apply structured design techniques to all of the design domains. By using the techniques in the top level system design down through the detailed hardware and software designs, it is hoped that fewer design anomalies will result. This paper will first review the flight test experiences of three highly complex, integrated aircraft programs: the X-29 forward-swept wing, the advanced fighter technology integration (AFTI) F-16, and the highly maneuverable aircraft technology (HiMAT) program. Significant operating anomalies, and the design errors which cause them, will be examined to help identify what functions a system design/information tool should provide to assist designers in avoiding errors.

Nomenclature

AFTI	advanced fighter technology integration
AI	artificial intelligence
DFD	data flow diagram
FCC	flight control computer
FCS	flight control system
FMEA	failure modes and effects analysis
HARV	high-angle-of-attack research vehicle
HiMAT	highly maneuverable aircraft technology
H/W	hardware
ILS	instrument landing system
KB	knowledge base
KBS	knowledge-based system
KCS	knowledge capture system
KEE™	Knowlege Engineering Environment (trademark of Intellicorp, Mountain View, CA)
KR	knowledge representation
LE	leading edge
NWS	nosewheel steering
PLC	power lever control
S/W	software
TE	trailing edge
T/O	takeoff
V and V	verification and validation

Background

System engineering has been recognized as an essential element in the development of complex systems. The top-down structured approaches to system engineering have been slow to catch on because of a lack of computerized

*Aerospace Engineer.
†Staff Engineer.

tools. However, recent advances in personal computers and new software (S/W) tools have reestablished the use of these structured system engineering methods.[1]

The system design aspects of the system design/ information tool expand on the current systems engineering methods by 1) automatically creating a knowledge base (KB) of the processes, data flows, and externals; and 2) including functions to verify consistency in design requirements unique to flight crucial control systems.

The information aspects of this tool address the need to provide design and implementation information throughout a flight control system's (FCS's) life cycle, and, specifically, to the test engineers. The verification and validation (V and V) effort for the digital FCS is of particular concern to the test engineer. Complete V and V is required to assure flight safety and requires the design information to establish, run, and analyze the V and V tests. Problems associated with V and V have caused major digital FCS developments to slip by as much as 18 months.[2] The system design/information tool needs to include the flight control design knowledge and its hardware (H/W) and S/W implementation.

Figure 1 shows a typical life cycle for an FCS and how the system design/information tool would support all phases. Shown is a typical life cycle for an FCS and how the life cycle phases relate to the system/information tool's capabilities. Some of the current tools which would share information with the system design/information tool are shown in the lower half of the figure.

The following review of research aircraft and the unique design errors that were found shows how system complexity can hide design errors from even the most experienced engineers. These errors reflect, in part, the difficulty of adequately communicating the system design details to the test engineers in the multiple disciplines. These disciplines include flight control law development, H/W design and test; S/W specification, coding and test; system integration and test; and flight test operations.

X-29 Description and Airdata Single-Point Failure

The X-29A technology demonstrator aircraft is an experimental vehicle which integrates a number of advanced technologies. These technologies include a forward-swept wing, tailored composite wing structure, and full authority digital flight control. The aircraft is also highly unstable and is dependent on the triplex digital FCS for stability and handling qualities.[3]

The FCS feedback gains are scheduled using air data. Air data errors can cause incorrect flight control gains and loss of the aircraft. To avoid incorrect gains, the X-29A has three sources of air data. Redundancy management S/W takes the three air data values, detects any failures, and selects a value to be used in the control law calculations.

After flying over 200 flights, a serious design error in the redundancy management logic was found during verification of a new release of flight S/W being tested in ground-based simulation. The error was attributed to the multidisciplinary nature of the system and had been in the flight S/W since the 38th flight. A lack of detailed understanding about the interactions between the air data system, redundancy management S/W, and the flight control laws allowed for the design error to occur and is discussed below.

The fault detection level in the redundancy management S/W was set at a large value because of errors, such as position errors, possible between the probes at certain flight conditions (Fig. 2). In the case of a probe failure, air data errors as large as the fault detection level were allowed to pass through to the control laws. At the lower and slower end of the flight envelope, a fail-to-zero of the nose probe would not be detected. Simulations have shown that this single-point failure would change the gains to the point that the aircraft would become unstable and depart. For over 162 flights the aircraft was at risk of being lost because of a single air data failure. The system requirement was that the aircraft be operational after two air data failures. Until a subsequent software release corrected the problem, the aircraft was grounded.

AFTI F-16 Description and Flight 44 Anomaly

The advanced fighter technology integration F-16 program investigated the integration of emerging technologies into an advanced fighter aircraft. The AFTI's three major technologies investigated were (1) flight-crucial digital control, (2) decoupled aircraft flight control, and (3) integration of avionics flight control and pilots display.[2]

The AFTI F-16 flight control system was a triplex, asynchronous digital system. The asynchronous architecture meant that input signals from sensors and controllers were read at different times into the three computers using a high-speed serial, digital data link (Fig. 3). Concerns for S/W reliability were addressed with the inclusion of a triplex, analog-independent backup unit.

The following summarizes an in-flight anomaly which occurred on flight 44.[2] This anomaly was the result of the interaction of many design characteristics and a unique flight condition. The characteristics included asynchronous computer operation, forward integrators in the control laws, and output redundancy management S/W. These characteristics coupled with a unique flight condition and resulted in the divergence of the three computers' output commands to the control surfaces. The redundancy management S/W in each of the channels declared the other two channels as failed. The pilots indication of this apparent simultaneous failure of all three computers was a dual fail flight control light in the cockpit. The end result of this in-flight anomaly was that the aircraft safely landed on what was effectively a single-string flight control system, even though no actual H/W failure had occurred.

Like the X-29 example, the AFTI F-16 had a serious design error resulting from the lack of a detailed understanding of the interactions between the many different disciplinary areas. In this case the design error was not recognized until after an in-flight anomaly was experienced.

HiMAT Design and Gear Deployment Anomaly

The HiMAT demonstrator was a remotely piloted research vehicle which incorporated such advances as composite structures, aeroelastic tailoring, reduced static stability, and digital flight control.[4] The aircraft was remotely piloted because the technologies represented too high a risk for a manned vehicle.

The HiMAT was flown remotely with the pilot in a ground-based cockpit and the control laws calculated in ground-based computers. Surface commands were telemetered to the aircraft as were aircraft sensor data which were telemetered to the ground (Fig. 4). The onboard digital flight control computers were dual redundant and processed uplink and downlink data. In the case of a complete loss of the dual uplink commands, the onboard system acted as a backup FCS capable of orbiting the aircraft until control was reestablished.

An anomaly occurred during the flight test program which resulted in the aircraft landing with its landing skids retracted. However, the pilot performed an excellent landing and the aircraft was not seriously damaged. The anomaly was induced by a single failure in the redundant uplink H/W. The onboard redundancy management S/W identified the failure and allowed for continued control of the aircraft, except for the deployment of the landing skids.

The anomaly was caused by a timing change made in the ground-based system and the onboard S/W for uplinking the gear deployment command. This change coupled with the onboard failure of one uplink receiver to cause the anomaly. The timing change was thoroughly tested with the onboard flight S/W for unfailed conditions. However, the flight S/W operated differently when an uplink failure was present. This critical information about the S/W was not readily available to the flight test team.

Requirements for a System Design/Information Tool

These brief examples demonstrate that system complexity is overwhelming the individual's ability to understand the entire system and the interactions that can take place between the different functional areas. The X-29 example illustrates how important the air data system and the redundancy management S/W design information is to the flight control designers and test engineers. Using the experience gained from the above aeronautics projects, we have formulated the requirements for a system design/information tool. The requirements include:

1. A system design capability to ease the capture of design information. The system design capability will provide a graphical, structured method for designing complex systems. It will help the designer avoid errors and allow the capture of the design information as it is created. Later in the aircraft development this information, in the form of an intelligent documentation system, will provide information to the test engineers.

2. Online documentation of all the information describing an FCS and the relationships between different disciplinary information. This includes H/W, S/W, redundancy management, and flight control law disciplines. The test engineer can then easily and graphically see the design information needed to qualify the system, thus avoiding the in-flight consequences of design errors.

3. Expert system functions to help analyze the relationships between the disciplines and uncover where unwanted interactions can occur. These functions can be used by designers, as well as test engineers, to assess the system's operations and avoid serious design errors.

4. Ability to perform failure modes and effects analysis (FMEA) on the many design iterations. Currently, FMEA is only performed on the H/W, not on the system as a whole. Because of the time required to perform an FMEA, the FMEA is usually performed once and is done with an early design iteration. The inability to analyze the final design raises questions of the FMEA's value. Automated FMEA using the current online design is one example of a capability that would assist designers and test engineers in finding serious design errors in a timely manner.

5. Links from the system requirements to the S/W and H/W designs. The links will allow the system requirements to be verified against the proposed implementation. Verification could then be done in an automated fashion, prior to committing to the build phase. This rapid prototyping concept would increase the chance of finding serious design errors prior to flight test.

Currently, some system design tools have become commercially available, but they do not address the needs of flight-crucial systems and only create conventional databases called data dictionaries. The actual H/W and S/W implementation information is not an integral part of these tools.

Description of the Knowledge-Based System Design/Information Tool

The following section will review the work accomplished to date and show how it applies to the larger problem outlined above. The methods for capturing system design knowledge, examples of what can be done using this knowledge, and an overview of the structure of the knowledge-based system (KBS) will be discussed. In related work, a good approach to design knowledge capture for the space station can be found in Wechsler.[5]

Focus

The effort is focused on the development of a generic knowledge capture system (KCS) for digital FCSs, which utilizes mature AI technology. The KCS is being used to capture design knowledge for the NASA high-angle-of-attack research vehicle (HARV), a modified F/A-18A with a thrust-vectoring capability. The primary efforts to date have been focused on the development of an intelligent documentation system for the system and H/W design realms. Examples of expert analyses that can be accomplished once the design knowledge is captured are described in the sections on the spin recovery system and nosewheel steering behavioral model.

The Knowledge Capture System

A major portion of the effort for the KCS has been devoted to the development of a knowledge representation (KR) which is tailored to the specific needs of the FCS problem domain. Four domains of knowledge have been identified within the FCS problem domain: system design, H/W design, S/W design, and utilities. These four domains provide the flight test engineer with the diverse kinds of information needed and the relationships which exist between them. Each domain possesses its own unique KR.

System Design Realm

The structured analysis methodology is used to describe the system design.[6] This methodology is based on a top-down hierarchical decomposition of system requirements using an extremely graphical user interface. The decomposition continues until the requirements are given with an adequate degree of detail. This design methodology creates a cleaner, more understandable design.

The tool creates linked hierarchical trees of data flows, processes, and externals. Each node in the process tree represents a process and is provided with a process description and other unique attributes which are stored in slots. To support flight control system design, the tool stores and tracks requirements for failure probability and mission criticality. In addition, external agencies and data flow objects are identified in the KB. All of this information is depicted graphically in data flow diagrams (DFDs). Figure 5 depicts a level 0 DFD.

The concepts of a process, an external, and a data flow, as defined by structured analysis, are identified here as graphical objects and individually represented as frames. The properties of the process, external, data store, and data flow objects are stored in the slots of the individual frames associated with each of these objects. The name of the process, failure probability, and data flow inputs are all examples of slots. The nature of the slot values can draw from the full spectrum of the paradigms supported by the Knowledge Engineering Environment (KEE™). Namely, they may be simple values, pointers to other frames, inherited values, active values, rules, and so forth. This KR will allow the users to perform various expert analyses of the system design. It is intended that the pointers, which are stored as slot values, will provide access to the related H/W, S/W, and utilities implementation knowledge stored elsewhere.

A hierarchical representation scheme is used for each of the three types of objects (processes, externals, and data flows). Each of these three hierarchies forms an individual, linked KB. In each case, the hierarchy is used to allow properties to be inherited and to identify the natural linkage between individual objects. These individual KBs are linked with pointers.

In a typical application of the structured analysis methodology, the data flow diagrams are viewed as an end object. In this KBS, the data flow diagrams are primarily viewed as a graphical front end for the KCS. Every data flow diagram image is mouse sensitive and possesses its own menu for entering and accessing knowledge. Now the test engineer can graphically see the relationships between systems, rather than trying to infer them from stacks of written text.

Hardware Design Realm

The knowledge representation for the H/W design is based upon the hierarchical block diagrams typically used in this problem domain. The nature of the representation is similar, although not identical, to the structured analysis methodology. The H/W objects are represented graphically as blocks, and these objects are decomposed in a hierarchical fashion until they have been described to an adequate degree of detail.

The connectivity between these H/W objects is indicated graphically with lines. These lines may represent various H/W connection abstractions such as data buses, a flow of information, or a form of control. In any case, this connectivity can be represented in the form of objects of a specific type and may possess a hierarchical characteristic.

The concepts of H/W and their connectivity are identified in the KBS as being objects and are individually represented as frames. The properties of these objects are stored in the slots of the individual frames. The nature of the slot values and the hierarchical relationship of the frame representation is the same as that provided for the system design. The H/W block diagrams serve as a graphical front end for the H/W design KB. Figure 6 depicts a top level H/W block diagram.

Software Design Realm

The structured analysis methodology is also used to describe the S/W design. The long-range plans include placing the KCS on a network with a workstation that has the flight code. This would give the KCS access to the flight code so that it would be possible to pull up listings of the flight code relevant to the objects defined in the S/W design data flow diagrams.

Utilities Design Realm

The utilities consist of the electrical power, hydraulic power, and environmental control systems which form an infrastructure for the FCS, and any embedded avionics system for that matter. The KR for this realm will utilize the structured analysis methodology to encode the design knowledge. It will also utilize the block diagram representation described previously for the H/W design realm. This representation will be used to encode the implementation knowledge.

Authoring and Browsing Mechanisms

The authoring mechanisms allow the user to create, delete, connect, and locate the user interface graphical images with mouse and menu commands. These ordinary capabilities provide typical graphical interface features. More importantly, the authoring mechanisms allow the user to *properly* create, delete, rename, and connect objects in the semantic network. The objects and their linkage within the semantic network are highly structured and canonical. A failure to conform to this formal structure would destroy the inheritance, browsing, and reasoning functionality. The authoring mechanisms also serve to enforce the methodologies which have been deemed appropriate for the individual realms within the KB.

The authoring mechanisms provide mouse and menu commands for inserting slot values for the properties of the objects in the KB. These mechanisms include an editor window for entering text descriptions and popup selection menus for properties whose slot values are restricted to a specific set of legal values. In some cases, the entry of slot values is monitored by demons. These demons actively monitor the knowledge entry and warn the user if the new value lies outside an envelope which is known to be valid. A demon is used to verify that failure probability requirements are kept as the design is decomposed.

The browsing mechanisms allow the user to display the text description, hierarchical relationships, and properties for the objects which have been entered into the semantic network. This information is accessed through the graphical, menu-driven, and mouse-sensitive user interface. This interface supports a random access to the KB. The knowledge representation supports a browsing strategy similar to the way we, as humans, pursue problem solutions. In this case, the network structure tends to guide the user in exploring the KB.

A Decomposition of the Spin Recovery System

The HARV flight tests will include research flight work with an angle of attack greater than 55°. For safety of flight in this regime, a spin chute recovery system has been added to the F/A-18A research aircraft. The following describes a decomposition which has been performed on the spin chute recovery system using the KCS.

Figures 7 and 8 show the hierarchical decomposition of the primary system that will deploy a spin chute for a spin recovery. These figures depict two of the many H/W diagrams involved in the decomposition of the spin chute recovery system. Figure 7 includes a partial display of the hierarchy. Figure 8 indicates the use of dual abstractions.

The box/line graphics provide an abstraction which follows directly from the top level H/W diagram graphical user interface. These box/line abstractions are linked directly to the objects in the hierarchical H/W design KB. The bit map graphical depiction of the circuit diagram has been added to clarify the user interface at the component level.

The two abstractions are tied together with bit-mapped hot spots. Authoring mechanisms allow the user to link the box/line abstractions (and correspondingly their H/W and signal flow objects) to as many hot spots on the bit map abstraction as may be desired. Browsing mechanisms allow the user to mouse on a hot spot or a box/line abstraction. Highlighting indicates the correspondence between a selected hot spot and its box/line abstractions. Similarly, highlighting indicates the correspondence between a selected H/W object (or signal flow object) and the relevant hot spots.

Further work in this area will be concentrating on the ability to take detailed H/W diagrams and perform failure modes and effects analysis of the systems they represent.

A Behavioral Model for the Nosewheel Steering System

The KBS includes dynamic behavioral models to aid the test engineer and also provides for rapid prototyping of the system. These models will be used to describe the system operation as a function of its operating modes. The models will permit the user to interactively enter mode commands and explore their impact on FCS operation.

The behavioral model described here permits the user to issue cockpit mode commands to a nosewheel steering (NWS) model which indicates the response and changes in the aircraft state. The model is based upon a rule set and a forward chaining paradigm. A dynamic display of the rules and their execution is available to dynamically document the system operation.

The NWS is a secondary control system within the FCS which is only operable on the ground. It provides nosewheel angular deflection proportional to pedal force when engaged. There are three modes of operation: off, low gain, and high gain. The desired mode is selected by the pilot, with switches located on the control stick grip, and is a function of several inputs, such as wing fold and weight on wheels. The NWS switch is used for NWS engagement and mode control, while the autopilot switch is used for NWS disengagement on the ground.

A dynamic interactive display (Fig. 9) is provided to control and display the control stick switch commands, the NWS system status, the NWS system block diagram, and the relevant F/A-18A aircraft status. The display window of the control stick switches includes a control stick and KEE™ active images for the NWS and autopilot switches. This window, which is mouse sensitive, will accept switch commands in an identical fashion to those issued by the pilot by way of the actual aircraft control stick. The KEE™ active images, which depict the NWS switch and the autopilot switch, are mouse sensitive. It is possible to issue a momentarily depressed, held depressed, or released command with these images.

The display of the aircraft status is also mouse sensitive. It is possible to explore the NWS logical operation as a function of aircraft power, touchdown status, wing fold status, and launch bar status by mousing the appropriate active image. As these parameters are changed, the appropriate operational mode is dynamically updated and displayed in the F/A-18A operation mode window. The NWS-related aircraft operational modes are: power off, wings folded, taxi, takeoff (T/O), launch, in flight, and landing.

A rule-based system implements the NWS mode logic. These rules are activated by the control stick switch commands and by changes in the aircraft status variables. The NWS system response is displayed by highlighting the appropriate mode in the NWS control mode status window. The NWS system response is also displayed by highlighting the control path in the NWS system block diagram window. It is possible to trace the rule execution in a KEE™ dynamic forward chaining execution window. The rule displays are mouse sensitive and permit the user to display a selected rule.

Knowledge-Based System Structure

The KBS is coded in Common Lisp, utilizes an expert system shell called the knowledge engineering environment, and is currently under development on a Symbolics machine (Fig. 10). This rapid prototyping environment has been utilized for the development of a KCS, which is tailored to the needs of the FCS problem domain. The KCS can be ported to any platform which is supported by KEE™. These platforms currently include the Symbolics and other major computing environments.

The KCS includes authoring mechanisms which enable the user to build a semantic network uniquely appropriate to a particular FCS. The KCS also includes browsing mechanisms which provide access to the semantic network knowledge. Rule-based models perform their reasoning on the objects defined in the semantic network.

The semantic network is composed of four realms of knowledge: the FCS system design realm, the FCS H/W design realm, the FCS S/W design realm, and the FCS utilities design realm (Fig. 11). Each of these realms is implemented with linked hierarchical networks of objects. The KBS semantic network is formed by linking the hierarchical networks of the four realms. The objects are individually represented with a frame-based representation. Authoring mechanisms enable the user to define a semantic network of FCS objects and their properties.

The semantic network of FCS objects are defined in an environment which includes an inference engine. Reasoning functions are under development which will enable the user to view and analyze the objects defined. Three kinds of models are to be developed:

1. behavioral models
2. failure mode and effects models
3. fault tree analysis models

The behavioral models are to provide a dynamic presentation of how designated objects behave as a function of user commands, FCS state variables, and FCS modes. The failure mode models are to indicate the loss of functionality associated with component failures. The fault tree models are to provide a diagnostic capability for the loss of FCS functionality. This diagnostic capability will allow the user to identify the possible causes and to help isolate the actual cause of the loss of FCS functionality.

Concluding Remarks

This project has proven to be an ambitious one. The roughly three man-years of effort have yielded a prototype which promises to fulfill the objectives, stated earlier, for a useful flight-crucial segment of the high-angle-of-attack research vehicle flight control system.

So far, the promises of artificial intelligence have been fulfilled. It has been possible to develop a knowledge capture system that captures the flight control system knowledge in a form which is tailored to the problem domain and is accessible to the user in a friendly fashion. Furthermore, the modeling capability has proven the value of defining the flight control system objects in an environment with an inference engine.

The system and hardware design realms now have a working functionality. In the remaining one man-year of effort, it is projected that a working prototype, capturing knowledge in all four of the realms, will be implemented. In terms of computer resource requirements, the response time is generally adequate, and less than 10 megabytes on the hard disk have been required to date for the design knowledge.

Looking to the future, it is projected that this prototype provides an infrastructure upon which a full-scale, fully operational knowledge capture system will be built that includes design aid capabilities. Longer range visions include such growth possibilities as postflight fault diagnosis, realtime ground support of flight tests, and real-time monitoring in the cockpit. The complexity of today's advanced aircraft demand that tools such as this be developed and utilized throughout the life cycle to assure safe and efficient flight operations.

References

[1] Neubauer, Chris, Radloff, B., and Rhodes, D., *Optimizing Systems Engineering Teams, Methods, and Tools,* AIAA Paper 88-3870, October 1988.

[2] Mackall, Dale A., *Development and Flight Test Experiences With a Flight-Crucial Digital Control System*, NASA TP-2857, 1988.

[3] Chin, J., Chacon, V., and Gera, J., *X-29A Flight Control System Performance During Flight Test*, AIAA Paper 87-2878, September 1987.

[4] Evans, Martha B. and Schilling, L.J., *The Role of Simulation in the Development and Flight Test of the HiMAT Vehicle*, NASA TM-84912, April 1984.

[5] Wechsler, D.B. and Crouse, K.R., *An Approach to Design Knowledge Capture for the Space Station*, NASA TM-89272, 1987.

[6] DeMarco, T., *Structured Analysis and System Specification*, Prentice-Hall, 1978.

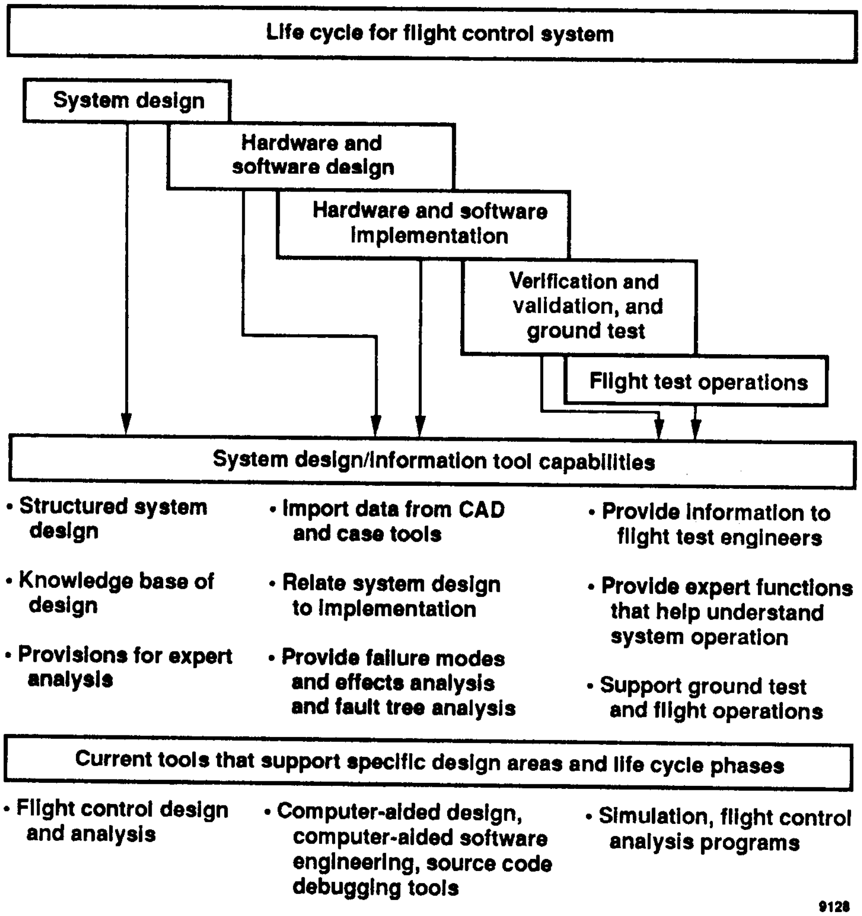
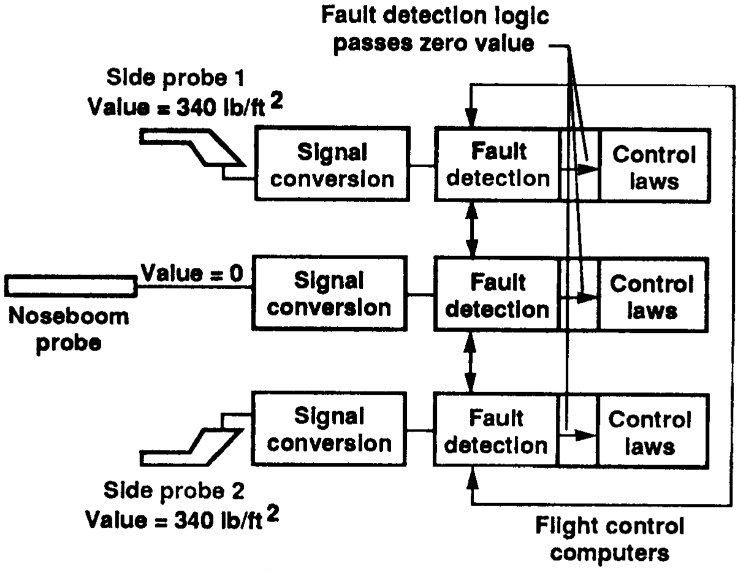

Life cycle for flight control system
System design
Hardware and software design
Hardware and software implementation
Verification and validation, and ground test
Flight test operations
System design/information tool capabilities
• Structured system design
• Knowledge base of design
• Provisions for expert analysis
• Import data from CAD and case tools
• Relate system design to implementation
• Provide failure modes and effects analysis and fault tree analysis
• Provide information to flight test engineers
• Provide expert functions that help understand system operation
• Support ground test and flight operations
Current tools that support specific design areas and life cycle phases
• Flight control design and analysis
• Computer-aided design, computer-aided software engineering, source code debugging tools
• Simulation, flight control analysis programs
9128

Fig. 1 Life cycle applications for the system design/information tool.



Hardware Diagram and Failure Condition

Fault Detection Logic

If noseboom value - side probe 1 < fault detection level and noseboom value - side probe 2 < fault detection level, then use noseboom value else declare noseboom failed and use average of side probes

Fault detection level = 354 lb/ft^2, 5 in. of mercury

9129

Fig. 2 Summary of X-29 failure condition and fault detection logic.

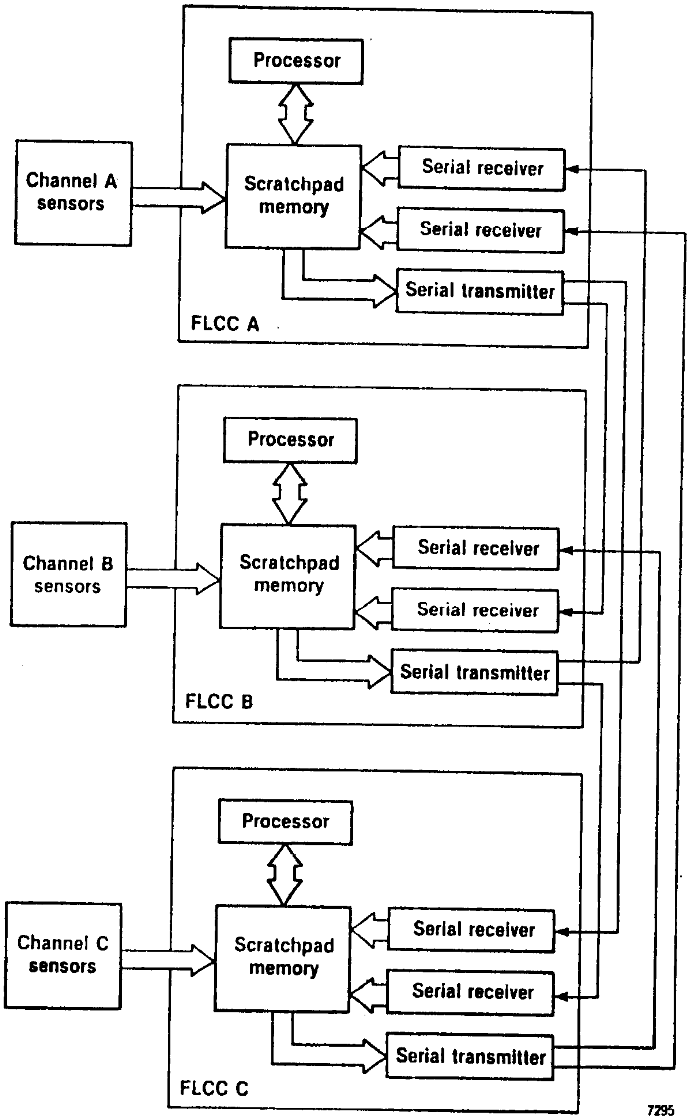


Fig. 3 AFTI F-16 cross-channel monitoring uses information sent on digital links.

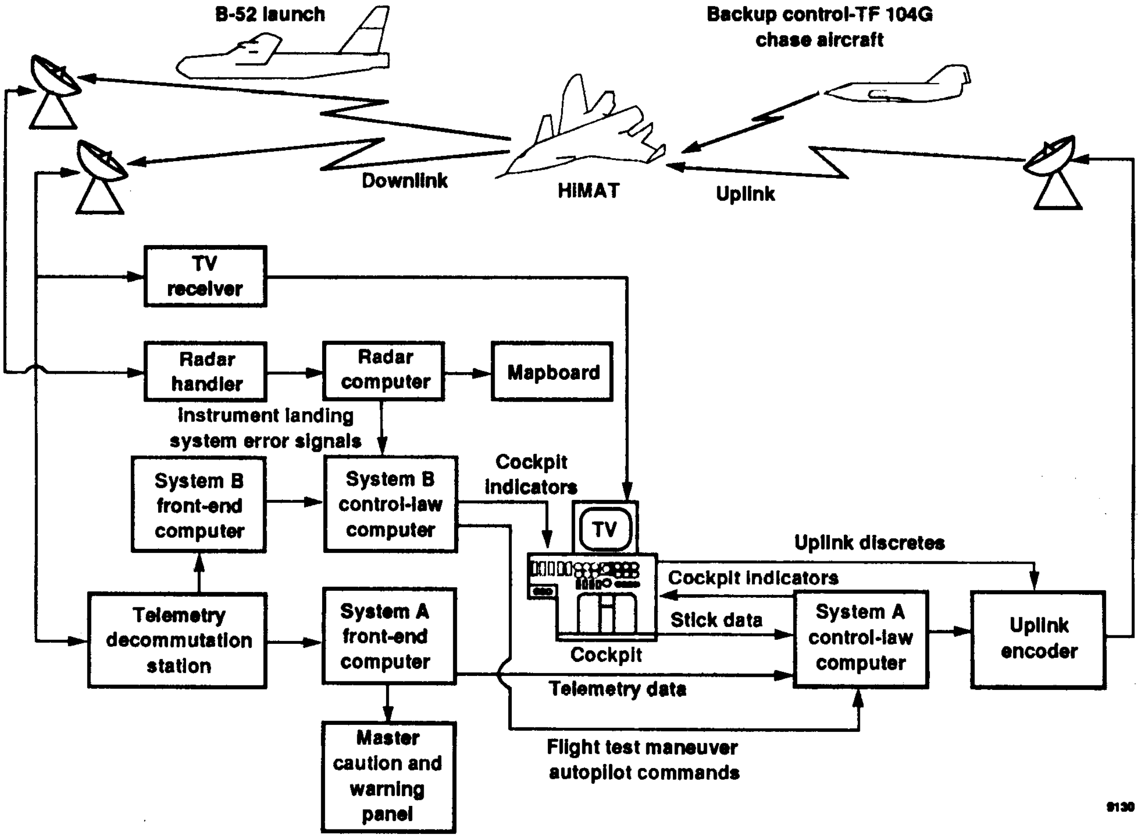

B-52 launch
Backup control-TF 104G chase aircraft
Downlink
HiMAT
Uplink
TV receiver
Radar handler
Radar computer
Mapboard
Instrument landing system error signals
System B front-end computer
System B control-law computer
Cockpit indicators
TV
Uplink discretes
Cockpit indicators
Stick data
Cockpit
Telemetry decommutation station
System A front-end computer
System A control-law computer
Uplink encoder
Telemetry data
Master caution and warning panel
Flight test maneuver autopilot commands
9130

Fig. 4 HiMAT control system.

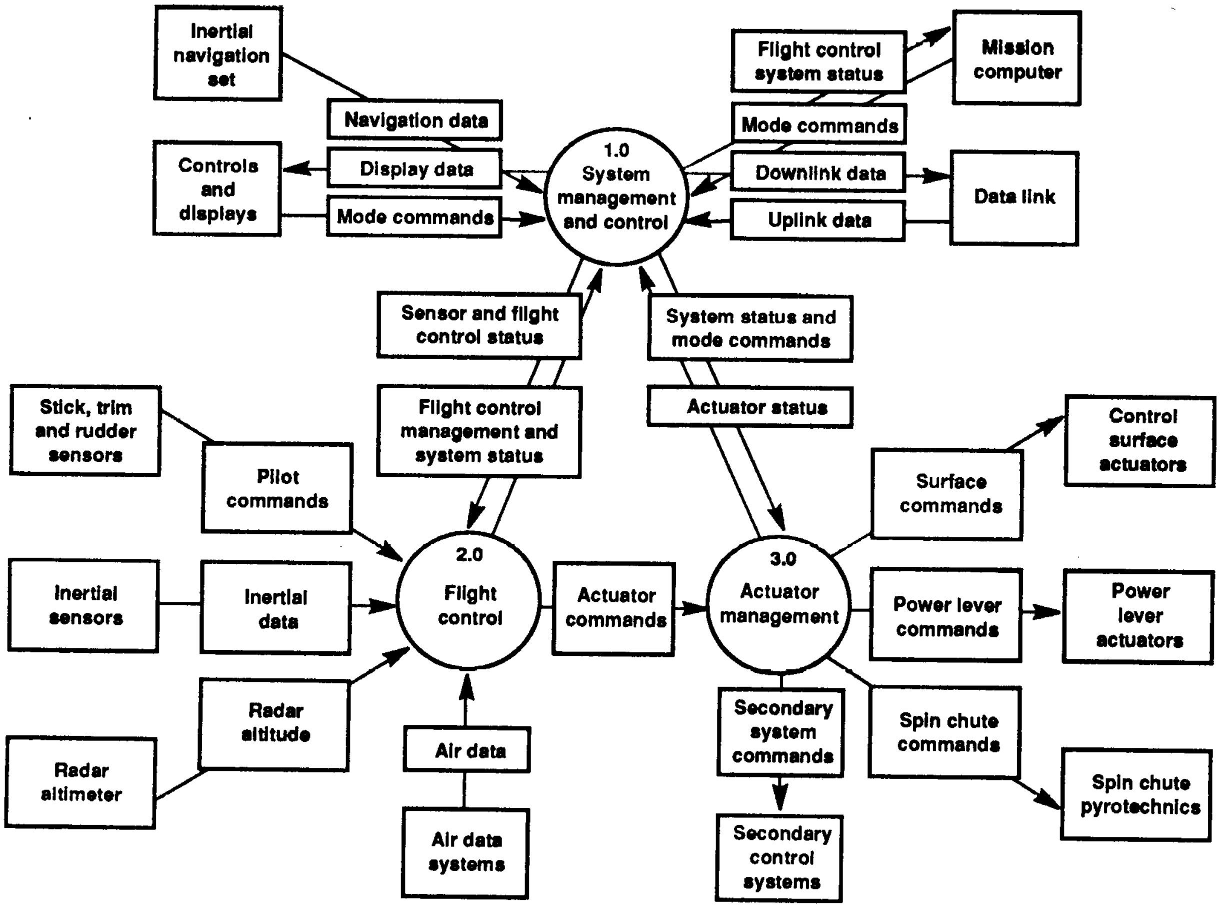


Fig. 5 F/A-18A flight control system—top level data flow diagram (level 0 DFD).

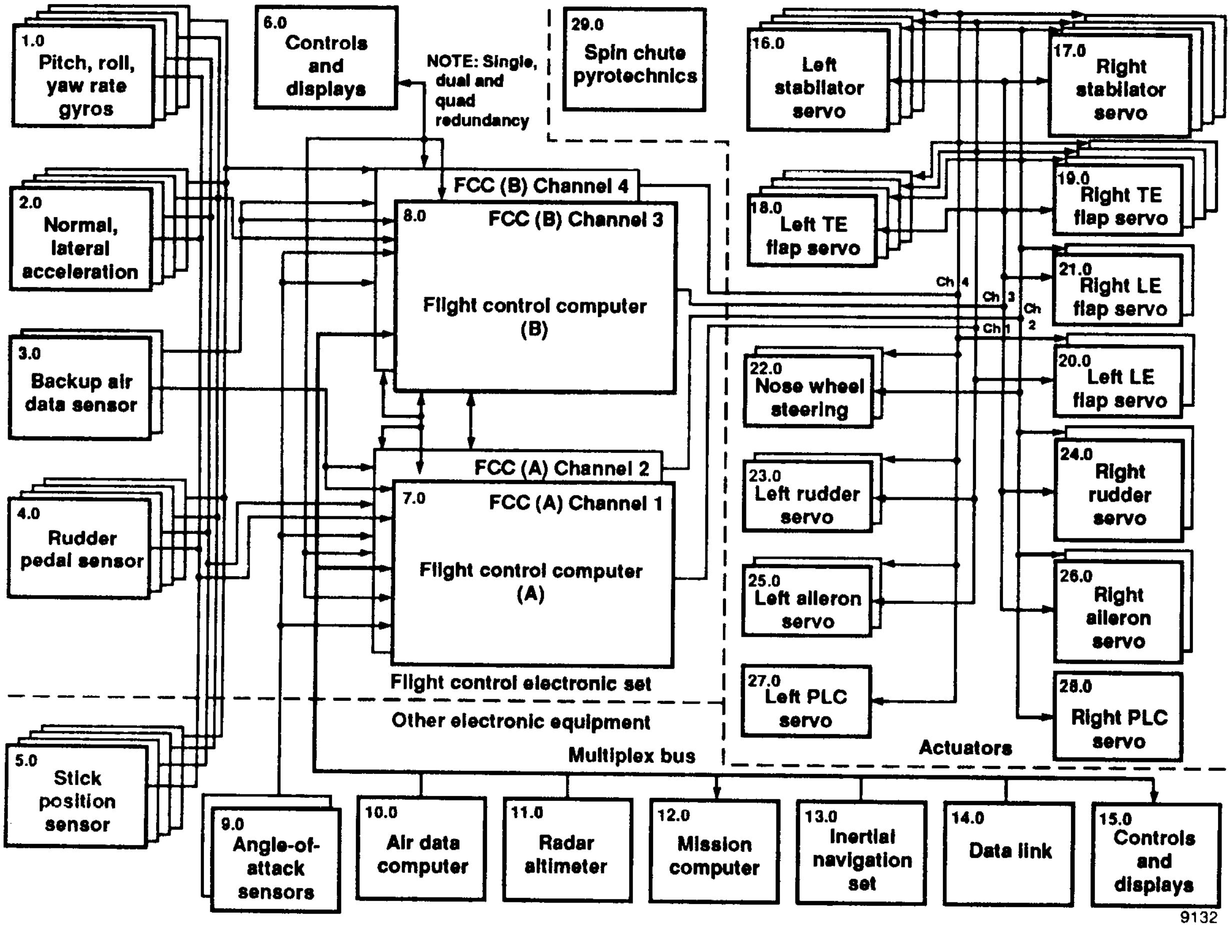


Fig. 6 F/A-18A FCS—top level hardware diagram.

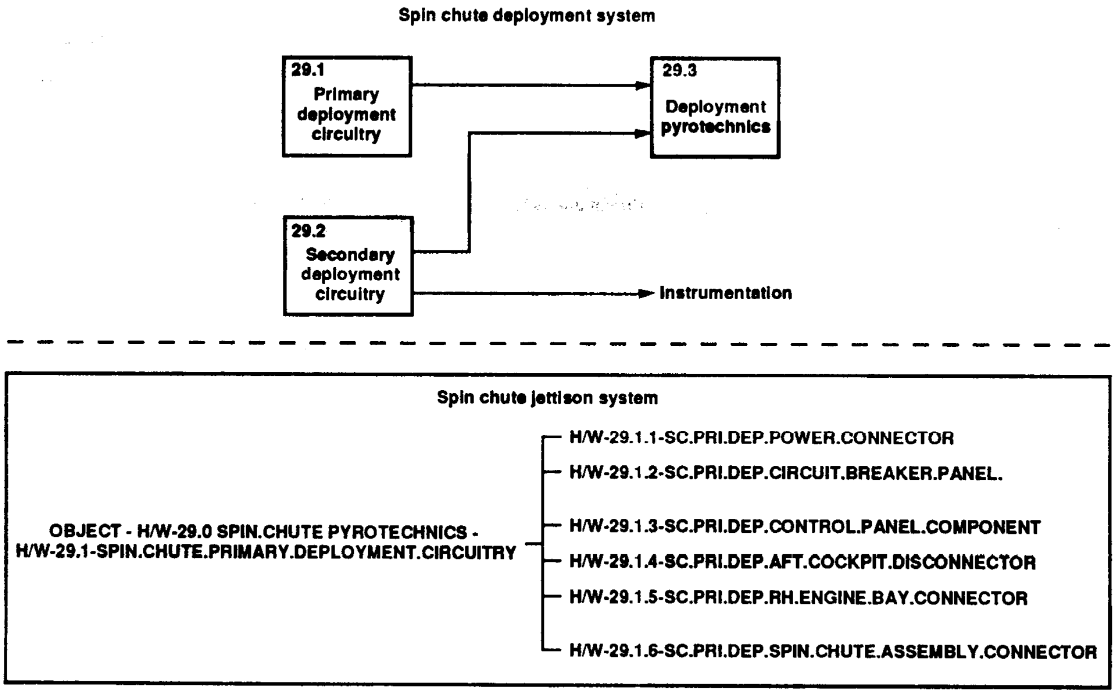

Spin chute deployment system
29.1
Primary deployment circuitry
29.3
Deployment pyrotechnics
29.2
Secondary deployment circuitry
Instrumentation
Spin chute jettison system
OBJECT - H/W-29.0 SPIN.CHUTE PYROTECHNICS -
H/W-29.1-SPIN.CHUTE.PRIMARY.DEPLOYMENT.CIRCUITRY
H/W-29.1.1-SC.PRI.DEP.POWER.CONNECTOR
H/W-29.1.2-SC.PRI.DEP.CIRCUIT.BREAKER.PANEL.
H/W-29.1.3-SC.PRI.DEP.CONTROL.PANEL.COMPONENT
H/W-29.1.4-SC.PRI.DEP.AFT.COCKPIT.DISCONNECTOR
H/W-29.1.5-SC.PRI.DEP.RH.ENGINE.BAY.CONNECTOR
H/W-29.1.6-SC.PRI.DEP.SPIN.CHUTE.ASSEMBLY.CONNECTOR

Fig. 7 F/A-18A flight control system—spin chute deployment system hardware diagram (level 1 HWD).

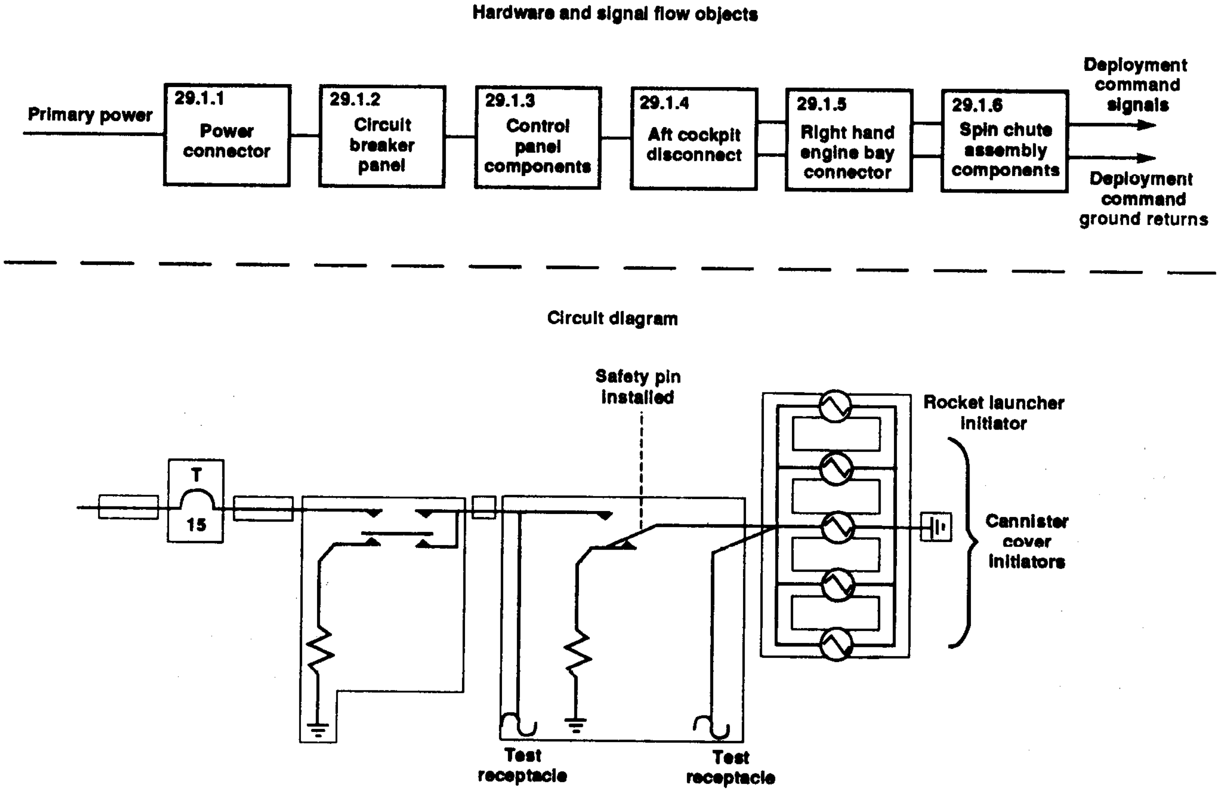

Hardware and signal flow objects
Primary power
29.1.1
Power connector
29.1.2
Circuit breaker panel
29.1.3
Control panel components
29.1.4
Aft cockpit disconnect
29.1.5
Right hand engine bay connector
29.1.6
Spin chute assembly components
Deployment command signals
Deployment command ground returns
Circuit diagram
Safety pin installed
Rocket launcher initiator
Cannister cover initiators
T
15
Test receptacle
Test receptacle

Fig. 8 F/A-18A flight control system—primary spin chute deployment circuit hardware diagram (level 2 HWD).

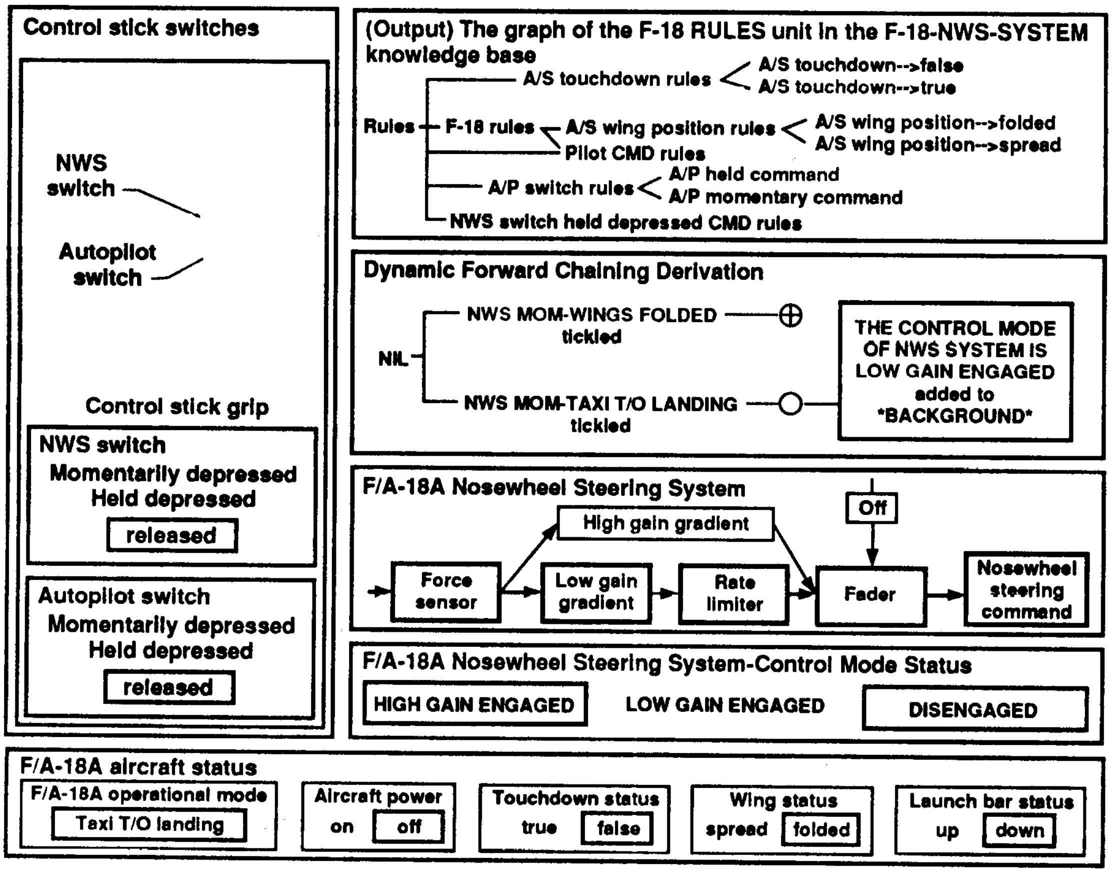


Fig. 9 F/A-18A FCS—nosewheel steering behavioral model display.

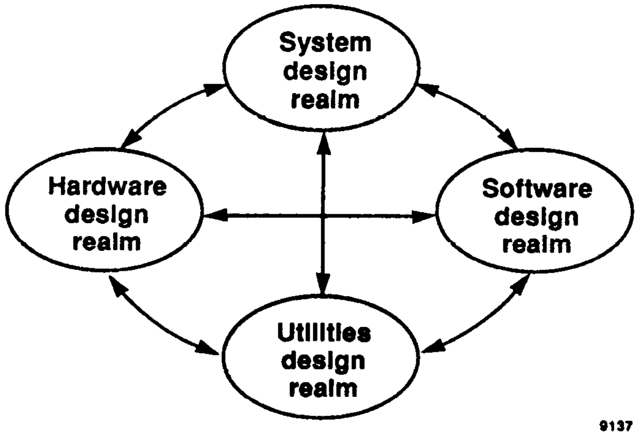


Fig. 10 The layered knowledge-based system architecture.

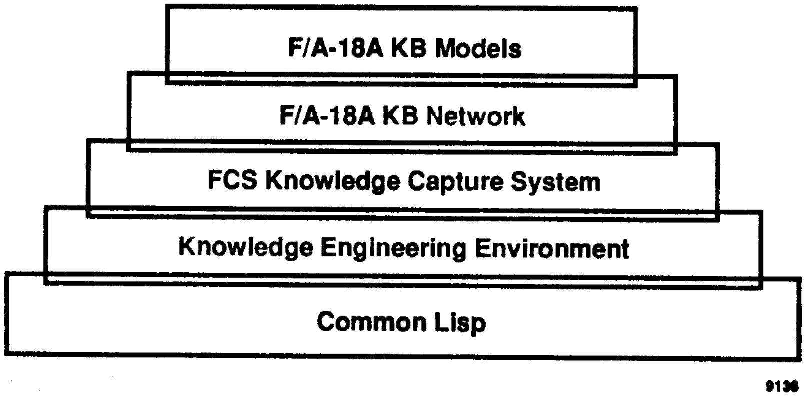

F/A-18A KB Models
F/A-18A KB Network
FCS Knowledge Capture System
Knowledge Engineering Environment
Common Lisp
9138

Fig. 11 The knowledge realms and their linkage.

NASA
National Aeronautics and Space Administration

Report Documentation Page

1. Report No. NASA TM-101704	2. Government Accession No.	3. Recipient's Catalog No.

4. Title and Subtitle

A Knowledge-Based System Design/Information Tool for Aircraft Flight Control Systems

5. Report Date
October 1989

6. Performing Organization Code

7. Author(s)

Dale A. Mackall and James G. Allen

8. Performing Organization Report No.
H-1546

10. Work Unit No.
RTOP 505-68-27

9. Performing Organization Name and Address

NASA Ames Research Center
Dryden Flight Research Facility
P.O. Box 273, Edwards, CA 93523-5000

11. Contract or Grant No.

12. Sponsoring Agency Name and Address

National Aeronautics and Space Administration
Washington, DC 20546

13. Type of Report and Period Covered
Technical Memorandum

14. Sponsoring Agency Code

15. Supplementary Notes

Prepared as AIAA 89-2978 for presentation at AIAA Computers in Aerospace Conference VII, Monterey, California, October 4-6, 1989.

16. Abstract

Research aircraft have become increasingly dependent on advanced electronic control systems to accomplish program goals. These aircraft are integrating multiple disciplines to improve performance and satisfy research objectives. This integration is being accomplished through electronic control systems. Because of the number of systems involved and the variety of engineering disciplines, systems design methods and information management have become essential to program success. The primary objective of the system design/information tool for aircraft flight control system is to help transfer flight control system design knowledge to the flight test community. By providing all of the design information and covering multiple disciplines in a structured, graphical manner, flight control systems can more easily be understood by the test engineers. This will provide the engineers with the information needed to thoroughly ground test the system and thereby reduce the likelihood of serious design errors surfacing in flight. The secondary objective is to apply structured design techniques to all of the design domains. By using the techniques in the top level system design down through the detailed hardware and software designs, it is hoped that fewer design anomalies will result. This paper will first review the flight test experiences of three highly complex, integrated aircraft programs: the X-29 forward-swept wing, the advanced fighter technology integration (AFTI) F-16, and the highly maneuverable aircraft technology (HiMAT) program. Significant operating anomalies, and the design errors which cause them, will be examined to help identify what functions a system design/information tool should provide to assist designers in avoiding errors.

17. Key Words (Suggested by Author(s))

Artificial intelligence, Causal modeling, Design knowledge capture, Flight control systems, Structured design

18. Distribution Statement

Unclassified — Unlimited

Subject category 62

19. Security Classif. (of this report) Unclassified	20. Security Classif. (of this page) Unclassified	21. No. of pages 19	22. Price A02

**For sale by the National Technical Information Service, Springfield, VA 22161-2171.*